6. THE PROGRAMMABLE INTERRUPT CONTROLLER

6.1 Interrupts in data processing

Most of the computer components need to exchange information with the microprocessor and wait for its attention when they require a data transfer. The microprocessor supervises the data transfer between different components in order to prevent the data losses. The microprocessor can supervise the data transfer in two different ways:

- by **Polling:** the microprocessor tests the devices one by one and serves the one that required special attention. Polling can be used in the case of some devices or in microprocessor systems, but not in PCs because it is too slow. Many processing cycles can be lost, because most of the times, the devices response is negative. In addition, the devices need data transfer or attention with different frequencies (e.g. the mouse needs much less attention than a hard-disk when it is activated for data transfer).
- by using **Interrupts:** it is another way of approaching the data transfer and it consists in letting the devices ask for attention when they need it, meanwhile the microprocessor can take care of other duties, wasting less time than in the polling mode. When an interrupt is sensed, the microprocessor quits the program it runs and properly serves the device that put it off.

The microprocessors generally have only one pin for the interrupt requests, but there can be more exterior interrupt sources. When the system needs more interrupts, an interrupt controller carrying out certain duties is placed between the interrupt sources and the microprocessor. Some of the duties carried out by the interrupt controller are:

- multiplexing the interrupts from different sources to the microprocessor pin
- solving the priority problems for simultaneous interrupts
- the generation of an interrupt vector which indicates the address of the program that handles the interrupt

The tasks above are solved differently for some microprocessors (e.g. Z80). The interrupt system is distributed to each circuit from the Z80 family and the interrupt priorities are solved depending on the circuit's position on a priority chain called "daisy-chain". The circuits also provide the corresponding interrupt vector. The PIO Z80 circuit from figure 6.1 has maximum priority, being the first component on the priority chain connected through the IEI (Interrupt Enable Input) and IEO (Interrupt Enable Output) signals. It should be pointed that ***in PCs, the PIC (Programmable Interrupt Controller) 18259A circuit is used***.

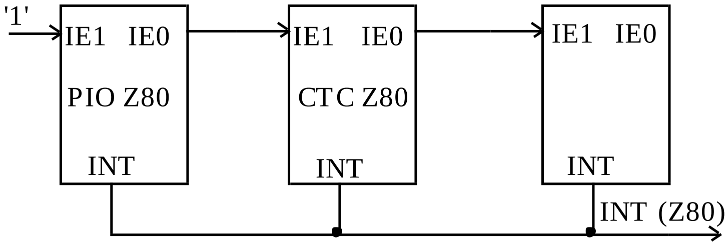


Figure 6.1. Priority Chain ('Daisy-Chain')

1. 8259A Circuit General Presentation

I8259A circuit is made in NMOS technology in a 28 pines capsule. This circuit is compatible with I8259 controller (used with I8080) and due to its additional functions it can be used in I80x86 microprocessor systems. The circuit allows interrupt requests active on increasing edge or on level high and it can be used in the system by himself, managing 8 interrupt levels, or more cascaded circuits can be used , to manage up to 64 interrupt levels.

Based on an interrupt handling rule that can be programmed, the controller finds out if there is at least one interrupt request and sets the INT signal on high level. If the microprocessor accepts the interrupt, it generates an interrupt accepting sequence of more /INTA run cycles (2 in I8088/I8086 mode and 3 in I8088 mode) to recognize the interrupt. During these cycles, the controller puts on the data bus the information required by the microprocessor for determining and executing the routine associated to the accepted interrupt level.

2.1. The I8259A Controller Architecture

The PIC block diagram has the following functional blocks:

Interrupt Request Register (IRR) – memorizes all the interrupt requests coming from outside.

In service Register (ISR) –memorizes in service interrupt requests at a certain time. After an interrupt has been served, the corresponding bit from ISR is invalidated. The reset can be done automatically or programmed.

Priority Comparator (PC) –compares the IRR register's content with ISR register's content and determines the interrupt request with greater priority than the interrupt being served, in which case it generates a new interrupt request by setting the corresponding bit in ISR during /INTA 's first cycle.

Interrupt Mask Register (IMR) – allows the invalidation of some interrupt levels by setting the corresponding bits in the register.

Data Bi-directional Amplifier –has 'three-state' bi-directional lines, connecting the controller's interface to the data bus**.** The control words, status information and vector type are transferred through the amplifier.

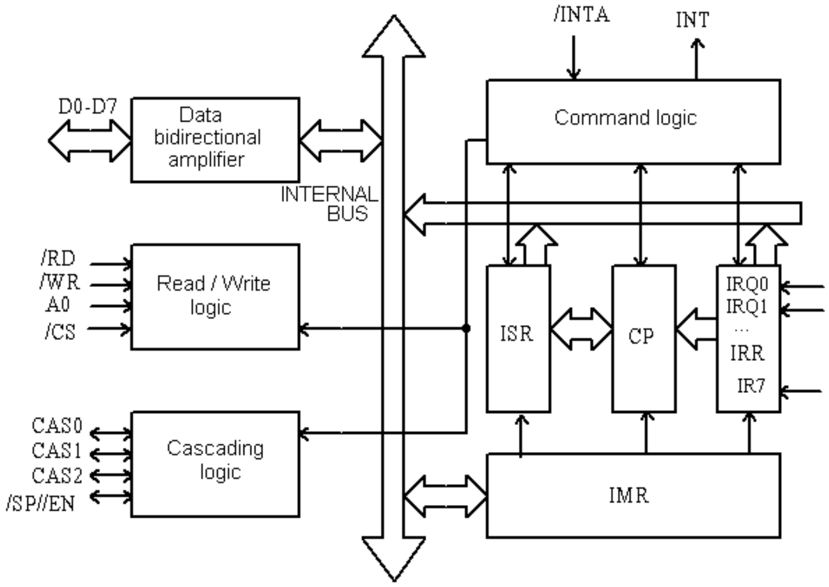


Figure 2. I8259A PIC Circuit Block Diagram

Cascading Logic

I8259A controllers can be cascaded to increase the number of interrupts; one controller is master and the rest of them slave (maximum 8 slave circuits). The slave controllers generate interrupt requests on the IRi master's input through the slave controller connected to that input. The CAS0-CAS2 signals of the master controller are output signals through which the slave controllers receive a code from the master circuit. The slave circuit which recognizes the code (the prioritary) will transmit, during /INTA cycles, the data for the accepted interrupt request, so that the routine can be determined and executed.

Read / Write Logic

The processor sees the controller as an input-output port set. Through A0, /WR, /CS and D0 – D7 signals, the processor programs the controller using command words, and through A0, /RD, /CS and D0 – D7 signals it reads the status registers.

The Command Logic ensures the interrupt request-acceptance protocol between the processor and the controller, through INT and /INTA signals.

2.2. Internal Registers

The address for the I8259A interrupt controller programming is 20H (for XT-PC), and the associated port addresses are listed bellow:

Port	I/O	Significance
20H	I	Read IRR , ISR
21H	I	Read IMR
20H	O	Write OCW2 if D4,D3 = 00
20H	O	Write OCW3 if D4= 1
20H	O	Write ICW1 , if D4 = 1
21H	O	Write OCW1, ICW2, ICW3, ICW4

The address for the slave PIC is 0A0H (in AT-PC)

Interrupt Request Register (IRR): (in al, 20H)

7	6	5	4	3	2	1	0
IRQ7	**IRQ6**	**IRQ5**	**IRQ4**	**IRQ3**	**IRQ2**	**IRQ1**	**IRQ0**

IRQi = 0 there was no interrupt request on line 'i'
= 1 interrupt request on line 'i'

In Service Register (ISR): (in al, 20H)

7	6	5	4	3	2	1	0
I	**-**	**-**	**-**	**-**	**W2**	**W1**	**W0**

I = 0 there is no interrupt
= 1 there is an interrupt

W2, W1, W0 contain the binary code of the most prioritary level which requested interrupt

Interrupt Mask Register (IMR): (in al, 20H)

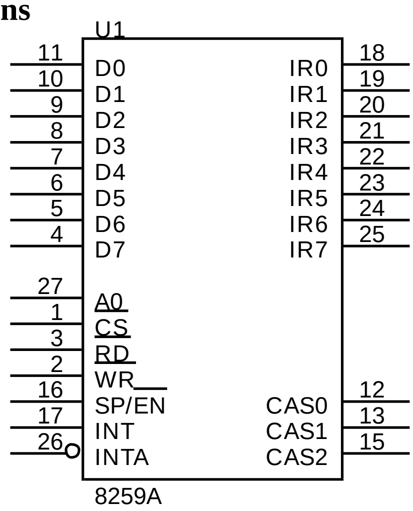
7	6	5	4	3	2	1	0
M7	**M6**	**M5**	**M4**	**M3**	**M2**	**M1**	**M0**

Mi = 0 IRQ1 interrupt line is not masked
= 1 IRQ1 line is masked

Priority Register (PR)

We do not have access to this register. The register compares the current interrupt's priority level with the one from ISR.

I8259A Circuit Pins



Name	**I/O**	**Pin**	**Function**
/CS	I	1	Circuit selection
/WR	I	2	Active on '0' level when command words are received from the microprocessor
/RD	I	3	Active on '0' level when reading I8259A circuit status
A0	O	27	A0 together with /CS, /RD, /WR determine the command / status word which the microprocessor writes / reads in / from I8259A circuit. Normally is connected to 0 address line (A1 for I8086)
D7 - D0	I/O	4 - 11	Bidirectional data lines for control, status information transfer and interrupt vector transfer.
CASO-CAS2	I/O	12-14	For cascading
/SP/ /EN	I/O	16	Slave Programme/ Enable Buffer is a double function pin. In buffered mode is used as a bus transfer control signal, and in un buffered mode, SP=1 for MASTER and SP=0 for SLAVE
INT	O	17	Interrupt – output connected to INT processor input; through it the interrupt request are transmitted
IR0-IR7	I	18-25	(Interrupt Request) - asynchronous inputs connected to the off-line circuits which generate interrupts to the processor, using PIC I8259A
/INTA	I	26	(Interrupt Acknowledge) - used for interrupt

			acknowledge and interrupt vector reading

2. PIC Programming

The circuit has only two ports for read or programme status, one must follow a certain order in sending the command words to the PIC. There are two types of words for working with I8259A:

- Initialization words ICW1, ICW2, ICW3, ICW4
- Operation words OCW1, OCW2, OCW3

The words sending order is presented in the diagram from figure 3. The ICW and OCW words structure is presented below:

ICW1 (OUT 20H, AL)

7	6	5	4	3	2	1	0
-	**-**	**-**	**1**	**LTIM**	**-**	**SNGL**	**ICW4**

The empty positions have significance in 8080 mode.
ICW4 = 0 ICW4 is not necessary
= 1 ICW4 is necessary
SNGL = 0 cascaded mode
= 1 single mode (only one 8259A circuit)
LTIM = 0 IRQ7 interrupt active on edge
= 1 interrupt active on level

ICW2 (OUT 21H, AL)

7	6	5	4	3	2	1	0
T7	**T6**	**T5**	**T4**	**T3**	**X**	**X**	**X**

Ti are bits 3 to 7 from the code sent in the second /INTA cycle. The number (type) of the interrupt vector for single mode is 8 (IRQ0), 9 (IRQ1), … F (IRQ7).

ICW3 (OUT 21H, AL) : for master

7	6	5	4	3	2	1	0
S7	**S6**	**S5**	**S4**	**S3**	**S2**	**S1**	**S0**

Si = 1 there is a slave connected to line ‘i’

ICW3 (OUT A1H, AL) : for slave

7	6	5	4	3	2	1	0
1	**x**	**X**	**x**	**x**	**ID2**	**ID1**	**ID0**

ID2, ID1, ID0: the identification code of the slave PIC (corresponds to IRQ line number from the master where it is connected). It is compared with the code emitted on CASO...2 by the master.

ICW4 (OUT 21H, AL)

7	6	5	4	3	2	1	0
0	**0**	**0**	**SFNM**	**BUF**	**M/S**	**AEOI**	**PROC**

PROC = 0 8080 mode, = 1 8086 mode
AEOI = auto EOI, = 0 manual EOI

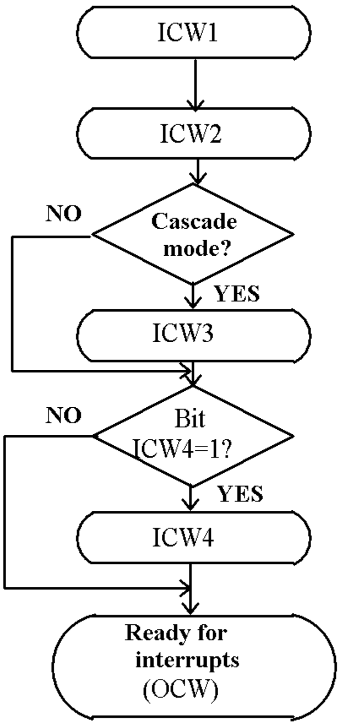
BUF	M/S	SIGNIFICANCE
0	X	No external data buffer
1	0	External data buffer; slave PIC
1	1	External data buffer; master PIC

SNFM (Special Fully Nested Mode)

= 0 Priority work mode. Only the prioritary interrupts are accepted (the interrupts with the same or less priority are ignored due to ISR)

= 1 Special mode for priorities (only for master). Other interrupts are accepted, regardless their priority; an interrupt with the same priority as the one being served will not be accepted

Figure 3. I8259A Controller Initialization Sequence



OCW1 (OUT 21H, AL) : for the interrupt mask

7	6	5	4	3	2	1	0
M7	**M6**	**M5**	**M4**	**M3**	**M2**	**M1**	**M0**

Mi = 0 unmasked level
= 1 masked level

OCW2 (OUT 20H, AL)

7	6	5	4	3	2	1	0
R	**SL**	**EOI**	**0**	**0**	**L2**	**L1**	**L0**

R (Rotate) = 1: until the next OCW2, the last served interrupt gets lowest priority
SL (Specific Level) = 1: specific EOI is used. See the table below for priority mode selection.
EOI (End of Interrupt) = 1: PIC is announced at the end of the served interrupt procedure
L2, L1, L0: contain the binary code of the served priority level

R	SL	EOI	Signification
0	0	0	Rotate in AEOI mode - Clear
0	0	1	Non-specific EOI
0	1	0	No effect
0	1	1	Specific EOI
1	0	0	Rotate in AEOI mode - Set
1	0	1	Rotate to non-specific EOI
1	1	0	Sets priority
1	1	1	Rotate to specific AEOI

OCW3 (OUT 20H,AL): masked mode, register read selection

7	6	5	4	3	2	1	0
-	**ESMM**	**SMM**	**0**	**1**	**P**	**PR**	**RIS**

ESMM (Enable Special Mask Mode) = 1: special mask mode validation
SMM (Special Mask Mode) = 1: special mask mode

ESMM	SMM	Significance
0	X	No effect
1	0	Special mask reset
1	1	Special mask setting

RP, RIS: select registers which will be read in the next instruction

RR	RIS	Significance
0	X	IMR selection
1	0	IRR selection
1	1	ISR selection

P = 0 polling mode (pseudo-interrupt mode: INT is not validated to the microprocessor. IRR register is swept through the program)
= 1 normal mode

3. I8259A controller modes

Fully Nested Mode

This mode is installed after the initialization if another mode is not programmed. The seventh interrupt level has the lowest priority and zero level becomes the most prior. During interrupt acknowledge cycle the most prior request is determined and also the vector associated to the interrupt is placed on the data bus. Meanwhile the corresponding bit from ISR is set. In EOI mode this bit stays in 1 and it needs an explicit command for erasing it before interrupt service routine ends. In AEOI mode the bit stays in 1 until the last interrupt acknowledge /INTA cycle ends. As long as the ISR bit is set, any other interrupt requests are ignored, except the more prioritary ones which are served (if the microprocessor is set to accept the interrupt requests).

EOI mode

The ISR bit corresponding to the served interrupt is automatically reset in AEOI mode (without using a special command) or using a command word activated before EOI mode interrupt serving routine ends. In cascaded mode the served interrupt corresponding bit has to be erased both from master and slave.

AEOI mode

Automatic End of Interrupt – is activated when AEOI bit from ICW4 is 1 and it is equivalent with a non-specific EOI command activated when the interrupt recognizing cycle ends (the third for I8080 and the second for I8086 mode).

Automatic (non-specific) priority rotation

There are cases when an off-line device after being served becomes the last one in the priority list and it won't be served until the other seven off-line devices from the priority list will be served properly.

Specific Priority Rotation

The user can dynamically change the priority order from the programme. A low priority level is defined and the other levels will be modified according to this one.

Interrupt masking

Each interrupt request input can be masked using IMR, which is programmed with OCW.

Special mask mode

In this mode the interrupt requests with the same priority as the served interrupt are invalidated but the more prioritary or less prioritary than the served one are accepted. This mode is programmed with ESMM = 1 and SMM = 1 in OCW3.

Polling mode

In this mode the interrupt system is invalidated at the microprocessor's level. The interrupt requests are scanned using a polling command. Polling mode is set with P = 1 in OCW3. The I8259A controller handles the first reading cycle (/RD and /CS) as the first cycle from the interrupt accepting sequence. The most prioritary request bit is set in ISR.

Specific Fully Nested Mode

This mode is necessary in the systems with several cascaded I8259A controllers and it is programmed from the master. (through SFNM = 1 in ICW4). This mode is similar to Fully Nested Mode with some exceptions. When an interrupt request coming from a slave is served, the master circuit does not invalidate this slave. In this way the more prioritary interrupt requests coming form the same slave are properly served. In the normal mode (FNM), when an interrupt request coming from a slave is served, the corresponding master input is invalidated and even the more prioritary interrupt requests than the one being served will not be accepted.

Buffer mode

When the I8259A controller is cascaded in the un buffered mode, the /SP//EN signal is used as an input signal controlling the type (master or slave). There are system configurations which need communication between the I8259A controller and the bus, using a bi-directional amplifier (buffer). In this case the /SP//EN signal is used as an output signal to establish the bi-directional amplifier's transfer direction:

/EN = 0 transfer from the controller to the microprocessor
/EN = 1 transfer from the microprocessor to the controller

Because of this, in the buffered mode a programme chooses the controller type. The third bit from ICW4 programs the buffered mode, and the second bit programs the master/slave type.

Cascaded Mode

One I8259A master controller can be interconnected with maximum 8 slave controllers to implement an interrupt system with 64 priority levels. The master controls the slave controllers through CAS0 … CAS2 cascading lines. The slaves' INT outputs, in cascaded configuration, are connected to the master's IRi inputs. If a slave's interrupt request is active and this interrupt request is the most prioritary one, the master will validate the corresponding slave through the cascading lines, so that the slave put the interrupt routine address or the interrupt type on the data bus.

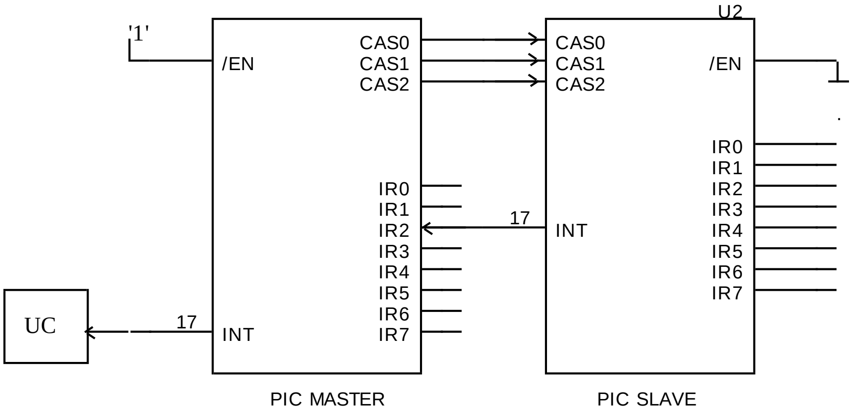


Figure 4. Controllers cascading in PC-AT

4. I8259A circuit utility in PC

In PC-XT computers only one I8259A controller is used, with 20h reference address and having the 8 inputs assigned to the off-line devices listed below:

Interrupt Input	Off-line device	Vector (type)
IRQ0 (most prioritary)	C0 Timer	08h
IRQ1	Keyboard	09h
IRQ2	Not used	0Ah
IRQ3	COM2	0Bh
IRQ4	COM1	0Ch
IRQ5	Hard Disk	0Dh
IRQ6	Floppy Disk	0Eh
IRQ7	LPT	0Fh

Table 1. PC-XT interrupts utilization

In PC-AT computers there are two cascaded controllers (see figure 3.), having 20h as master’s address and A0h for slave. The typical utilization and other possible utilization are listed in table 2.

IRQ	Priority	Vector	Typical Utility	Other Utilities
0	1	08h	Timer	-
1	2	09h	Keyboard	-
2	-	0Ah	Cascading (IRQ8-15)	Modem, EGA adapter, COM3/4
3	11	0Bh	COM2(serial port)	Modem, SB, COM4, Network Cards(NC)
4	12	0Ch	COM1	COM3, Modem, SB, NC
5	13	0Dh	Sound Blaster (SB), HD	LPT2/3, COM3/4, Modem, NC
6	14	0Eh	Floppy Disk Controller	Band accelerator
7	15	0Fh	LPT1(Parallel port)	LPT2, COM3/4, Modem, SB, NC
8	3	70h	Real time clock	-
9	4	71h	-	NC, SB, SCSI adapter, PCI device
10	5	72h	-	NC, SB, IDE channel, PCI device
11	6	73h	-	NC, SB, SCSI adapter, VGA

				card,…
12	7	74h	-PS/2 mouse	NC, SB, SCSI adapter, IDE channel
13	8	75h	Coprocessor (FPU)	-
14	9	76h	IDE-1 channel (HD)	SCSI adapter
15	10	77h	IDE-2 channel	SCSI adapter, NC

Non-masked Interrupt (NMI)

There is a separate pin for non-masked interrupt request, which is active on edge. The I bit setting does not influence the acceptance of NMI. The non-masked interrupt is used in PC when a parity error is detected.

5. Independent work

1. Study the architecture and programming mode for I8259A controller, and the utility in the IBM PC compatible computers of the controller.

2. Write a test program for PIC.

3. Where can we find in TVI the addresses of the interrupt handler routines, for the requests from the slave PIC inputs?

4. Write a program sequence, which masks the interrupts coming from different off-line devices (timer, keyboard, floppy disk), and notice the effect after running the program.

5. What is the role of the following sequence ?

```
mov al, 20h
out 20h, al
```

6. What is the IRET instruction run effect?

7. What involves the two I8259A controllers cascading, according to the connections, programming and priorities?